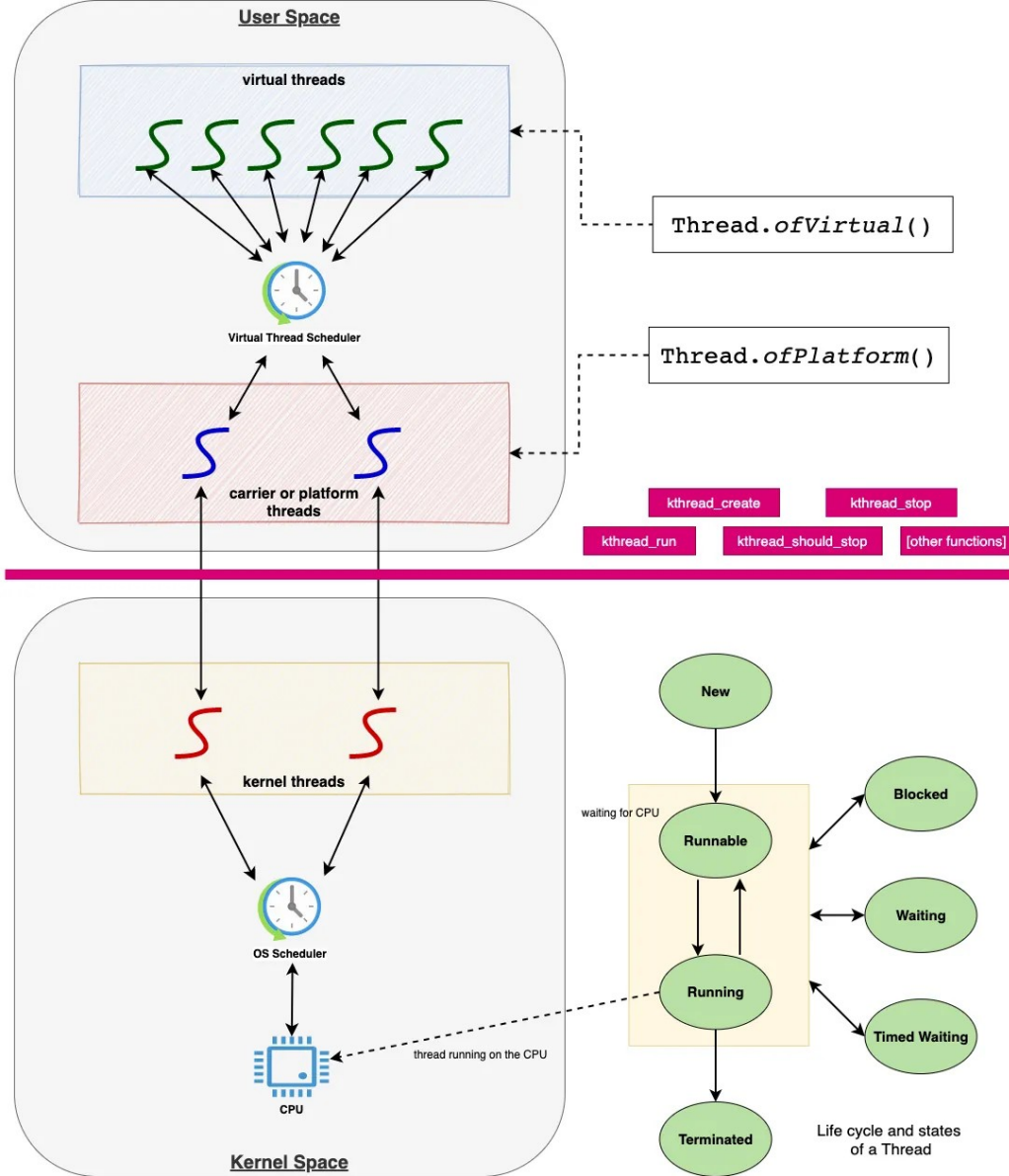


Virtual Threads

Virtual Threads 101

- Virtuální vlákna se chovají podobně jako go coroutines. X virtuálních vláken scheduler mapuje na platformová vlákna
- Jakmile je virtuální vlákno blokováno, čeká, nebo je uspané, uvolní platformové vlákno, které tím pádem může scheduler využít pro jiná virtuální vlákna
- Platform Thread:Kernel Thread = 1:1



Java Virtual Threads

- <https://openjdk.org/jeps/425>
- Na co se hodí a nehodí virtuální vlákna?
- Ideální použití:
 - I/O bound logika
 - Lepší utilizace zdrojů, alternativa pro reaktivní způsob programování
- Nevhodné použití:
 - CPU bound logika
 - Poolování virtuálních vláken (ale i to může někdy dávat smysl)

Spring Boot

- <https://www.infoq.com/news/2023/12/spring-boot-virtual-threads/>
 - `spring.threads.virtual.enabled=true`
- Virtuální vlákna se používají všude! (u tomcatu, hikari poolu, @Async metodách – tady pozor, tímto se mění dřívější výchozí chování!!!)
 - Pokud byste potřebovali omezit počet vláken @Async poolu:
 - <https://stackoverflow.com/questions/77825075/how-to-limit-the-number-of-spring-async-tasks-with-virtual-threads-enabled>
 - Nebo pomocí vlastního AsyncExecutoru, který bude mít omezenou velikost poolu.
 - Pozor na to, abyste měli všechny dependency včetně JDBC driveru aktualizované (kvůli potenciálnímu pinningu)
 - <https://github.com/brettwooldridge/HikariCP/issues/2151>

Virtual Threads pinning I.

- <https://mikemybytes.com/2024/02/28/curiosities-of-java-virtual-threads-pinning-with-synchronized/>
- <https://abhishekvrshny.medium.com/pinning-a-pitfall-to-avoid-when-using-virtual-threads-in-java-482c5eab78a3>
- A virtual thread cannot be unmounted during blocking operations when it is pinned to its carrier. A virtual thread is pinned in the following situations:
 - The virtual thread runs code inside a synchronized block or method
 - The virtual thread runs a native method or a foreign function (see Foreign Function and Memory API)
- Pinning does not make an application incorrect, but it might hinder its scalability. Try avoiding frequent and long-lived pinning by revising synchronized blocks or methods that run frequently and guarding potentially long I/O operations with `java.util.concurrent.locks.ReentrantLock`.

Virtual Threads pinning II.

- Jak odhalit pinning?

-Djdk.tracePinnedThreads=full

NEBO:

Pomocí JDK Flight Recorder Event Streaming:

- <https://openjdk.org/jeps/349>

5 věcí na co si dát pozor

- <https://quarkus.io/blog/virtual-thread-1/#five-things-you-need-to-know-before-using-virtual-threads-for-everything>

- Virtual Threads & Futures:
 - <https://blogs.oracle.com/javamagazine/post/virtual-threads-futures>
- Structured Concurrency:
 - <https://openjdk.org/jeps/453>